CS-565 Computer Vision

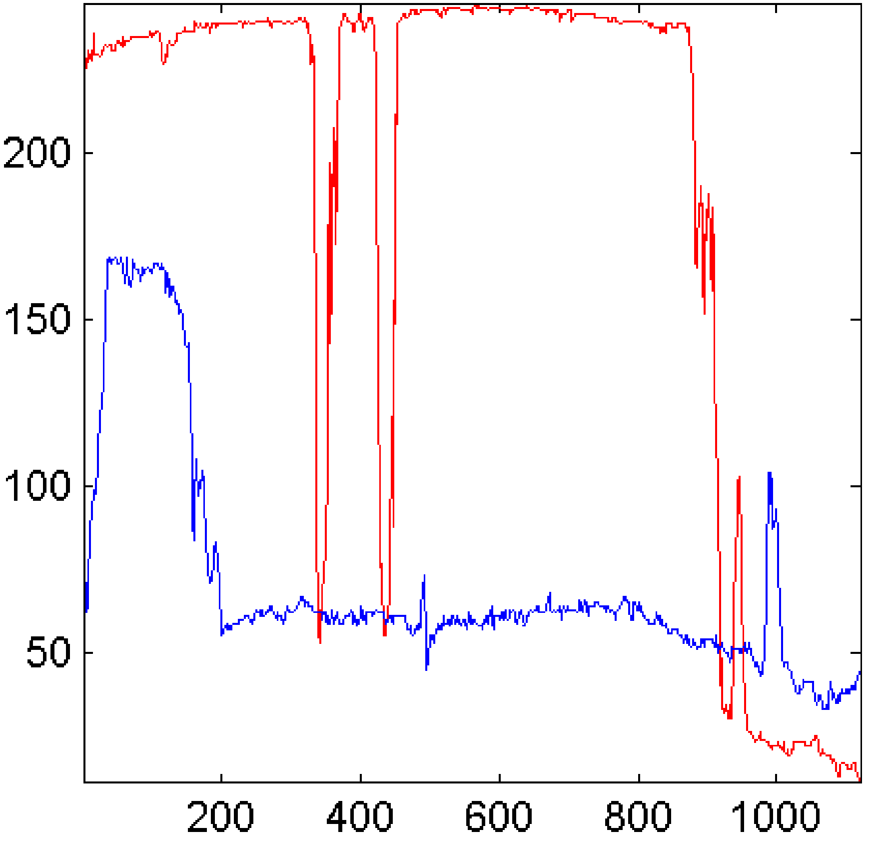
Nazar Khan
PUCIT

Lecture 9: Derivative Filtering and Edge Detection

Derivative Filtering

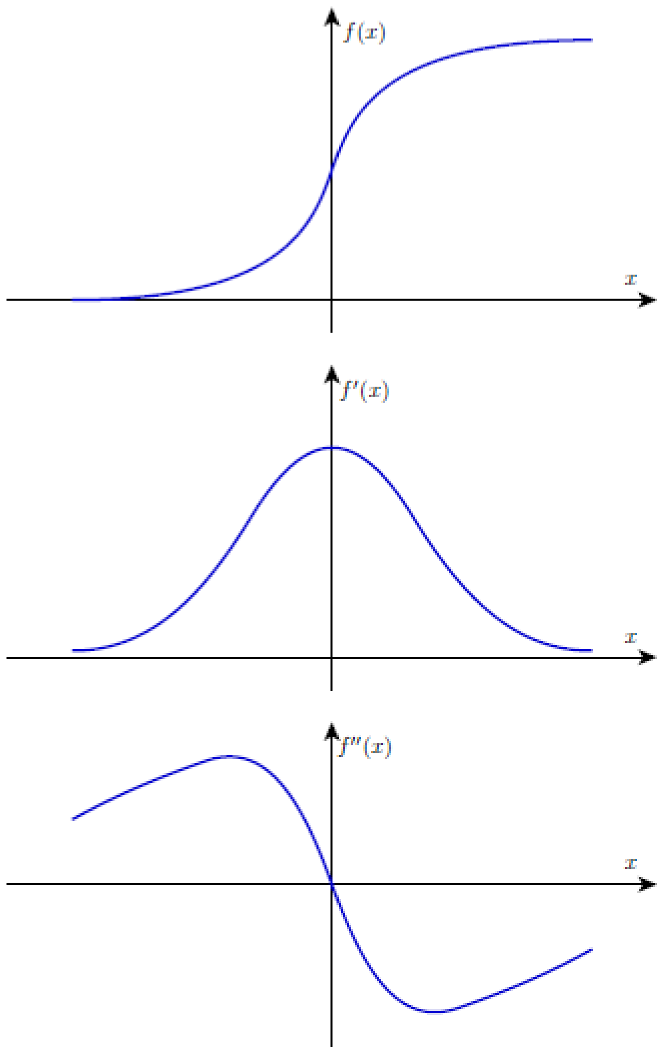
- Derivatives capture local gray value image changes.
  - Perceptually important features such as edges and corners.

Derivative Filtering



Left: A road image for which two 1D signals along the red and blue horizontal scan-lines have been extracted. **Right**: The intensity profiles along the red and blue scan-lines. The large jumps in the profiles correspond to boundaries of trees and road. Author: Nazar Khan (2014)

Derivative Filtering



1D signal

1st Derivative

Edge corresponds to maximum of 1st derivative magnitude.

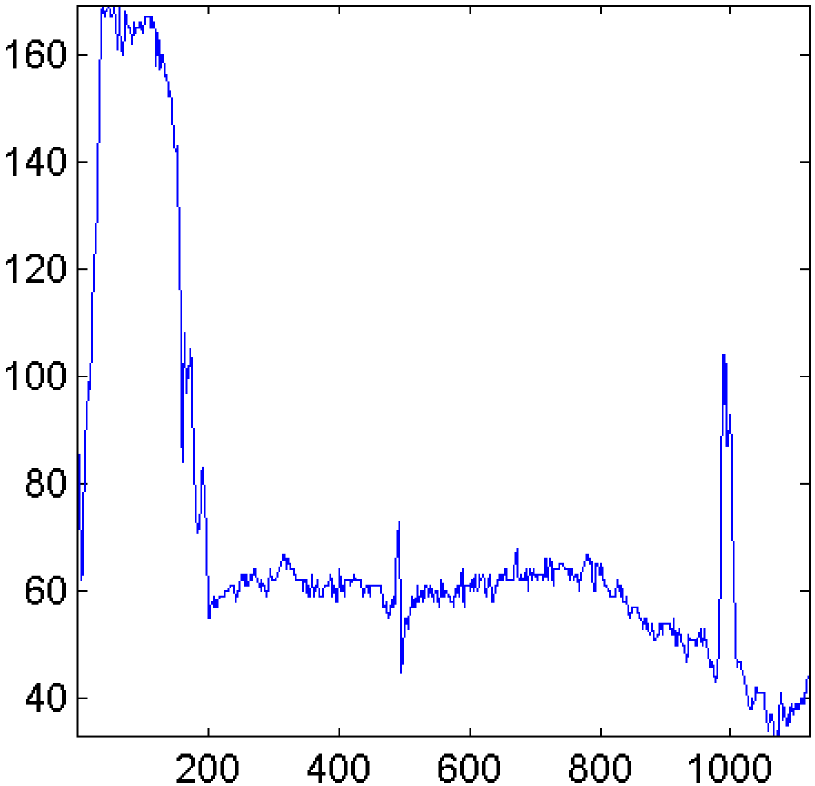
2nd Derivative

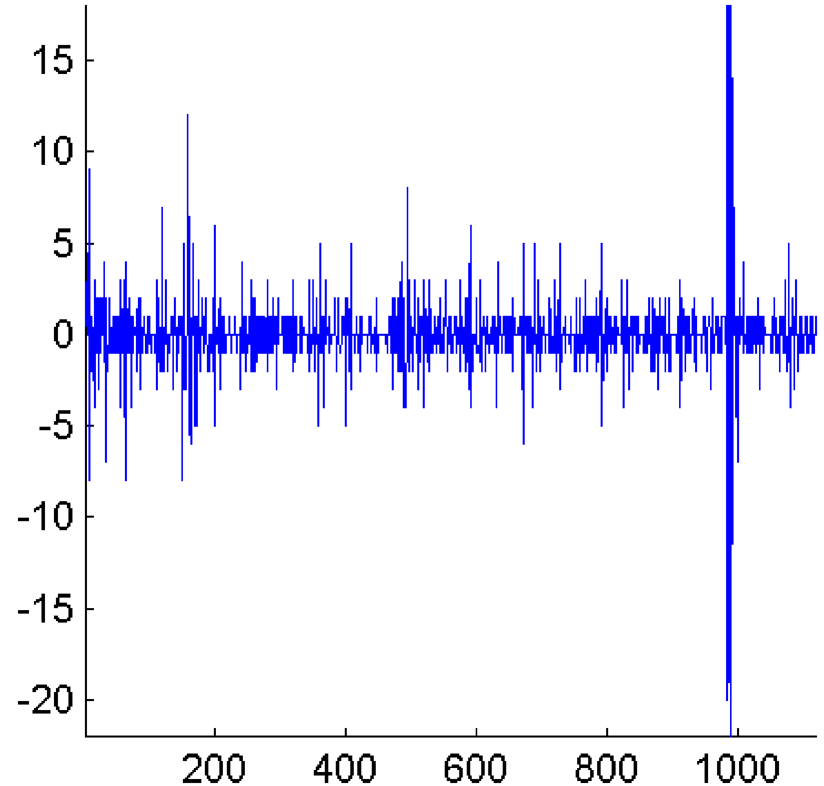
Edge corresponds to zero-crossing of 2nd derivative.

Derivative computation can be dangerous

- Noise can be amplified.
  - This leads to false edges.
- <u>Solution</u>: remove high frequencies before computing derivatives (often done through Gaussian convolution).

False Zero-crossings





Left: A 1D intensity profile. **Right**: Plot of the 2nd derivatives. The small preturbations (noise) in the input lead to false zero-crossings (edges) in 2nd derivatives. Author: Nazar Khan (2014)

Some Notation From 2D Calculus

- Let $f(x,y)$ be a 2D function.
- $\partial f/\partial x = \partial_x f = f_x$ is the **partial derivative** of f with respect to x while keeping y fixed.
  - $\partial^2 f/\partial x \partial y = \partial/\partial x\ (\partial f/\partial y)$
  - $f_{xy}=f_{yx}$ (under suitable smoothness assumptions).

Some Notation From 2D Calculus

- **Nabla operator** ∇ or gradient is the column vector of partial derivatives $\nabla=(\partial_x,\partial_y)$.
- $\nabla f=(\partial_x f,\partial_y f)$ points in the <u>direction of steepest ascent</u>
  - direction in xy space in which the function increases most rapidly.
- $|\nabla f|=\mathrm{sqrt}(f_x^2+f_y^2)$ is invariant under rotations.
- **Laplacian operator** $\Delta f = f_{xx}+f_{yy}$
  - Sum of 2nd derivatives

Continuous Derivative Approximation

- Using 2nd order Taylor's expansion

$$f(x+h) = f(x) + hf'(x) + h^2f''(x)/2 + O(h^3) \text{ ----- (i)}$$

$$f(x-h) = f(x) - hf'(x) + h^2f''(x)/2 + O(h^3) \text{ ----- (ii)}$$

- Subtracting (ii) from (i) and solving for f'(x) gives

$$f'(x) = (\ f(x+h) - f(x-h)\ ) / 2h + O(h^2)$$

- Adding (i) and (ii) and solving for f''(x) gives

$$f''(x) = (\ f(x+h) - 2f(x) + f(x-h)\ ) / 2h^2 + O(h)$$

- **(H.W) Prove that using a 1st order Taylor's expansion for f(x+h) gives**

$$\mathbf{f'(x) = (f(x+h)-f(x))/h + O(h)}$$

- **(H.W) Prove that using a 1st order Taylor's expansion for f(x-h) gives**

$$\mathbf{f'(x) = (f(x)-f(x-h))/h + O(h)}$$

Discrete Derivative Approximations

- Let h be the grid distance. For grids of image pixels, usually $h=1$.

f_{i-1}	f_i	f_{i+1}

- 1st derivative approximation
  - Forward difference $f_i'=(f_{i+1}-f_i)/h$
  - Backward difference $f_i'=(f_i-f_{i-1})/h$
  - Central difference $f_i'= (f_{i+1}-f_{i-1})/2h$
- 2nd derivative approximation
  - Central difference $f_i''= (f_{i+1}-2f_i+f_{i-1})/2h^2$

Masks
[**1** -1]
[1 -**1**]
½ [1 **0** -1]
½ [1 -2 1]

Discrete Derivative Approximations

- **Central difference approximations are preferred because of lower truncation errors**.
- Derivative filters are separable filters.
  - In 2D, use $\partial^2 f/\partial x \partial y = \partial/\partial x\, (\partial f/\partial y)$.
  - That is, convolve in one direction, then convolve the result in the other direction.
  - This way, we carry out 2D convolutions using 1D convolutions only which is cheaper (Assignment 1).

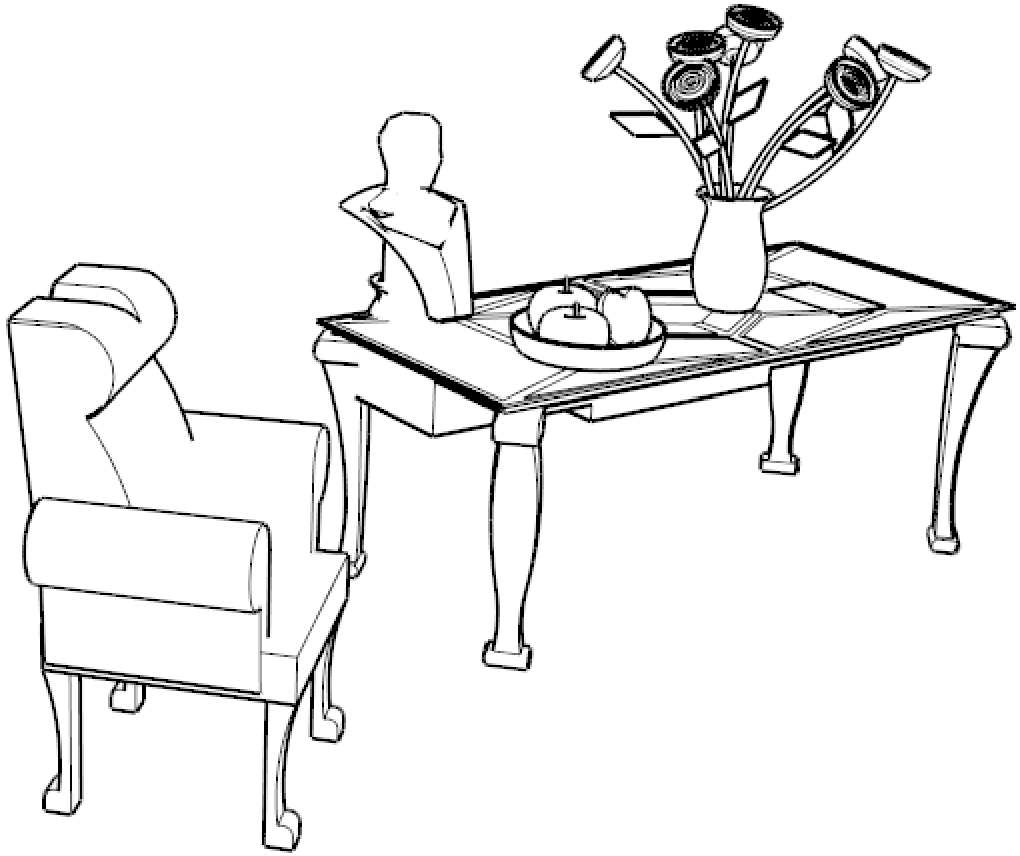
Average vs Derivative Filtering

- Average
  - All values are +ve
  - Sum to 1
  - Output on smooth region is unchanged
  - Blurs areas of high contrast
  - Larger mask -> more smoothing

- Derivative
  - Opposite signs
  - Sum to zero
  - Output on smooth region is zero
  - Gives high output in areas of high contrast
  - Larger mask -> more edges detected

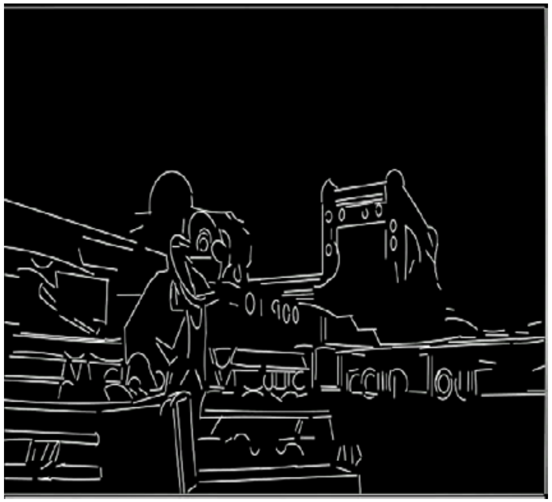
Edge Detection

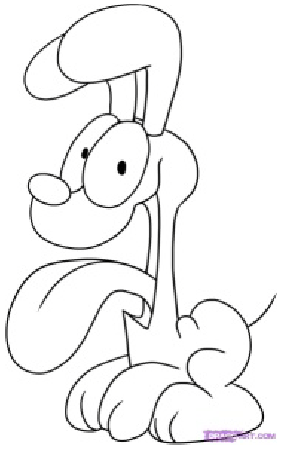
Edge Detection



Edge Detection

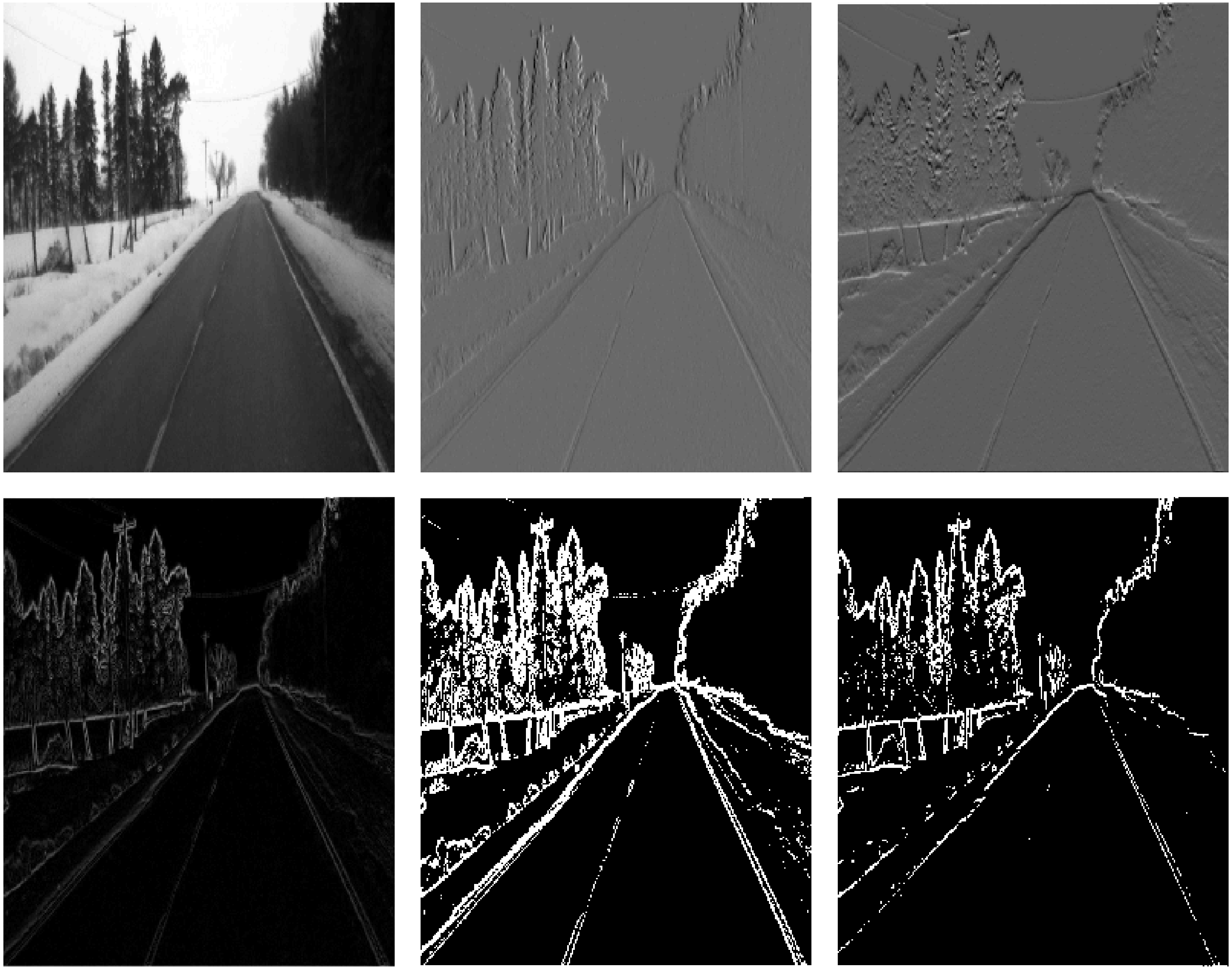
- Edge – strong change in local neighbourhood.
- One of the most important image features.
- We can understand comics and line drawings.
- Edge detection is the first step from low-level vision (pixel based descriptors) towards high-level vision (image content).
- Can be detected through 1st or 2nd order derivative operators.





Edge Detection via 1st Order Derivatives

1. Reduce high frequencies (noise) in image f by convolving with a Gaussian kernel K_σ. Smooth image $u = K_\sigma * f$.
2. Compute approximate 1st derivatives u_x and u_y and compute gradient magnitude $|\nabla u| = sqrt(u_x^2 + u_y^2)$
3. Edge pixels correspond to $|\nabla u| > T$.



Clockwise from top-left: Original image *I*, horizontal derivative image I_x, vertical derivative image I_y, gradient magnitude image $|\nabla u|$, $|\nabla u|>5$, $|\nabla u|>10$. Author: Nazar Khan (2014)

Edge Detection via 1st Order Derivatives

- Advantage
  - 1st order derivatives are more robust to noise than 2nd order derivatives.
- Disadvantages
  - Require threshold parameter T. Some edges may be below T.
  - Some edges may be too thick.
- Remarks
  - Suitable T strongly depends on σ. Why?
  - T can be selected as a suitable percentile of the histogram of $|\nabla u|$.
    - T is the 75th percentile if $|\nabla u| < T$ for 75% of the pixels.

Canny Edge Detector

- Among the best performing edge detectors.
- Mainly due to sophisticated post-processing.
- 3 main steps.
- 3 parameters
  - Standard deviation σ of Gaussian smoothing kernel.
  - Two thresholds T_{low} and T_{high}.

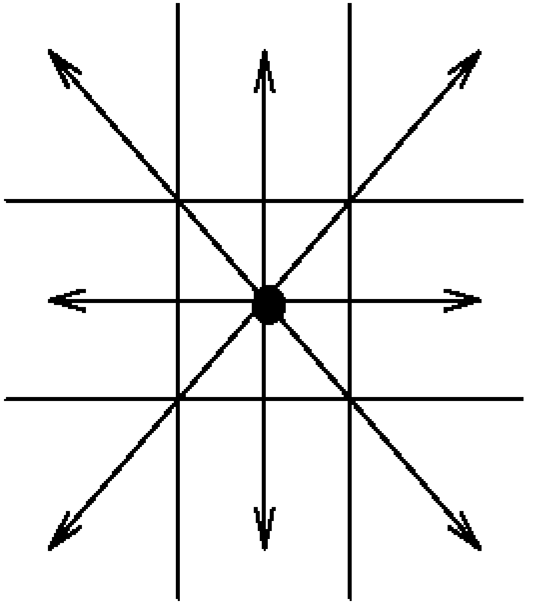
Canny Edge Detector

1. Gradient approximation by Gaussian derivatives
   – Magnitude $|\nabla u|=\mathrm{sqrt}(u_x^2+u_y^2)$
   – Orientation angle $\varphi=arg(|\nabla u|)=atan(u_x,u_y)$
   – $|\nabla u|>T_{low}$ are candidate edge pixels.
2. Non-maxima Suppression (thinning of edges to a width of 1 pixel)
   – In every edge candidate, consider the grid direction (out of 4 directions) that is "most orthogonal" to the edge.
   – Basic Idea: **If one of the two neighbours in this direction has a larger gradient magnitude,mark the central pixel**.
   – After passing through all candidates, remove marked pixels from the edge map.

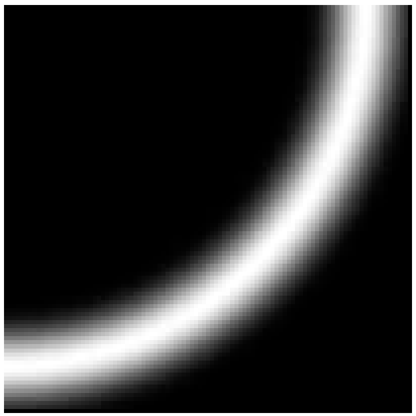
Non-maxima Suppression

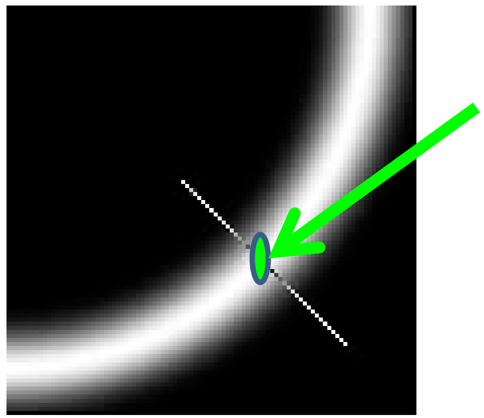
- Quantisation of gradient direction.

$$\theta = \arctan \frac{f_y}{f_x}$$



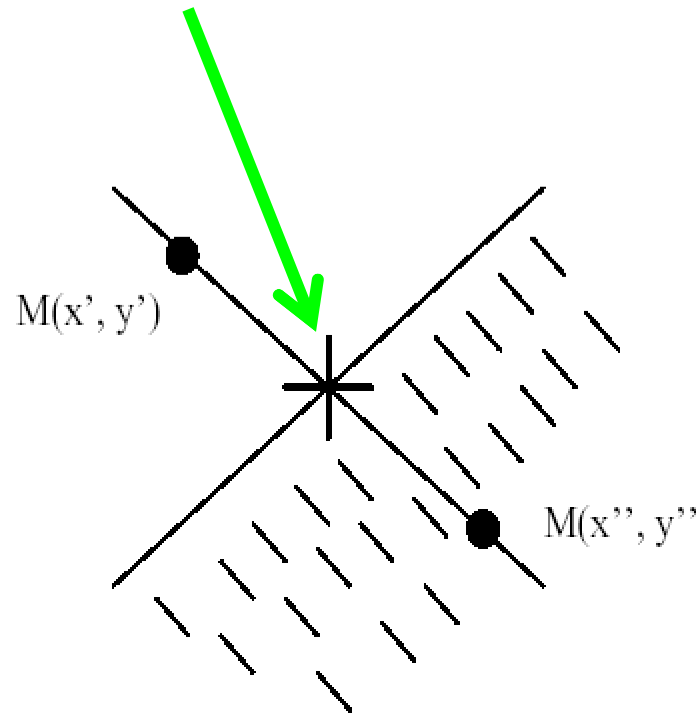
Non-maxima Suppression





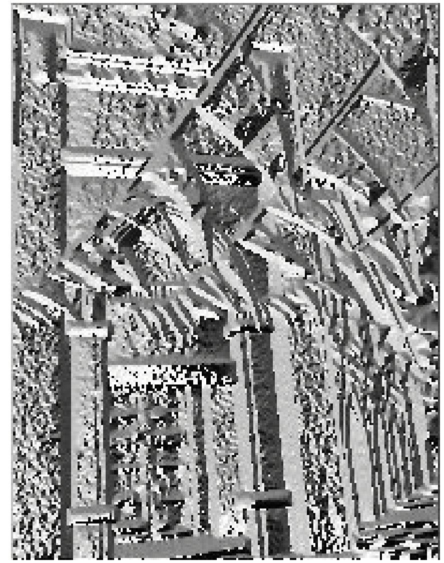
Only this point remains. The remaining magnitudes along the line are set to 0.

Remove all points along the gradient direction that are not maximum points.



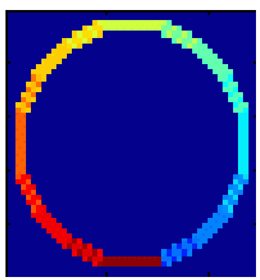
Original

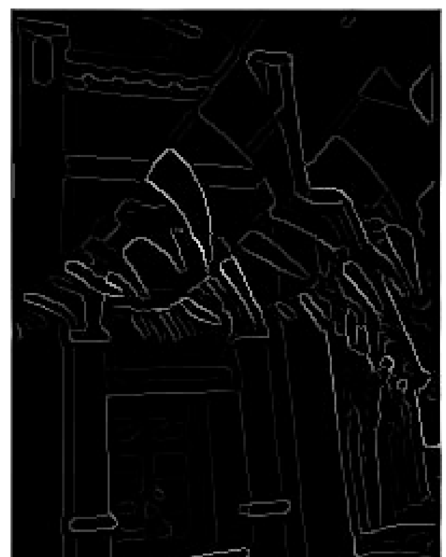
Gradient magnitude
$|\nabla u|$



Gradient direction φ

Quantized φ





Non-maxima suppression of $|\nabla u|$

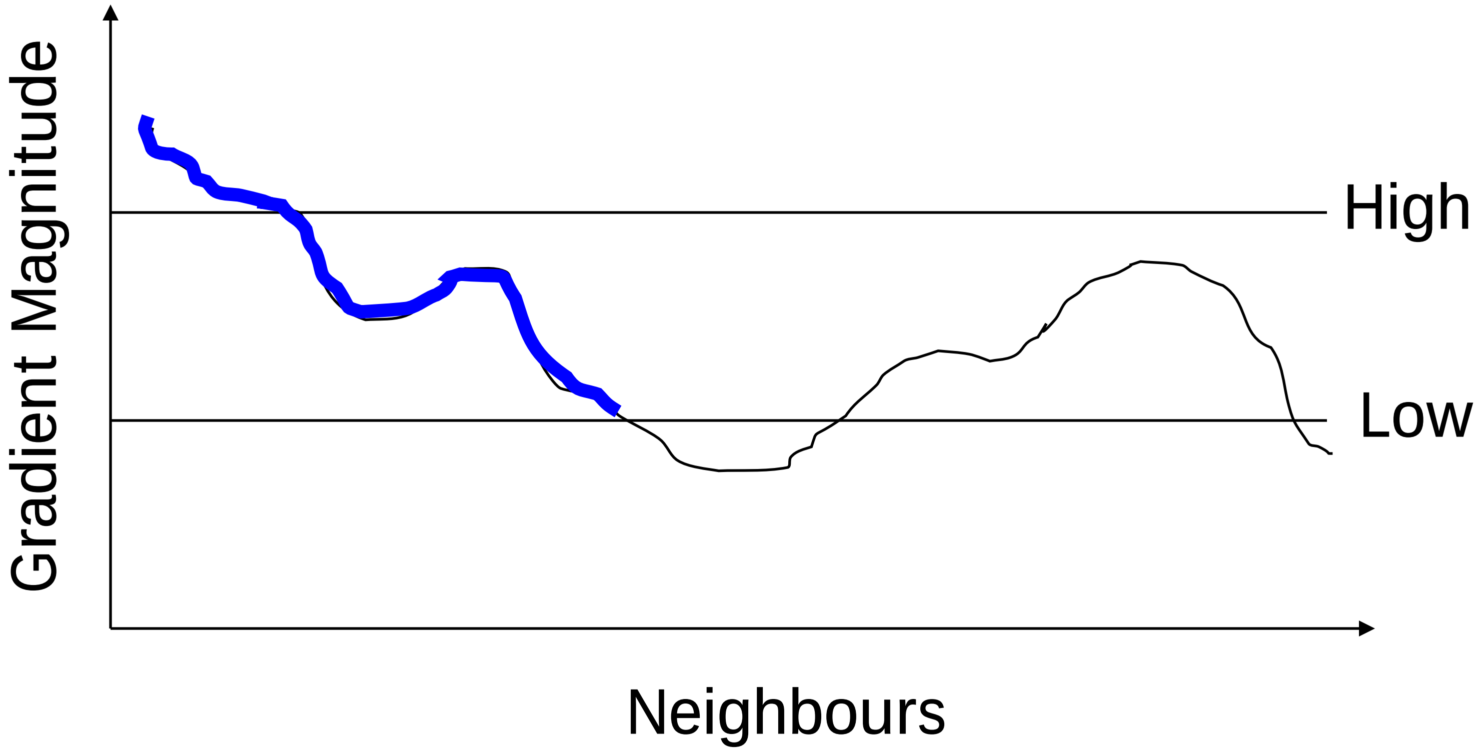
Canny Edge Detector

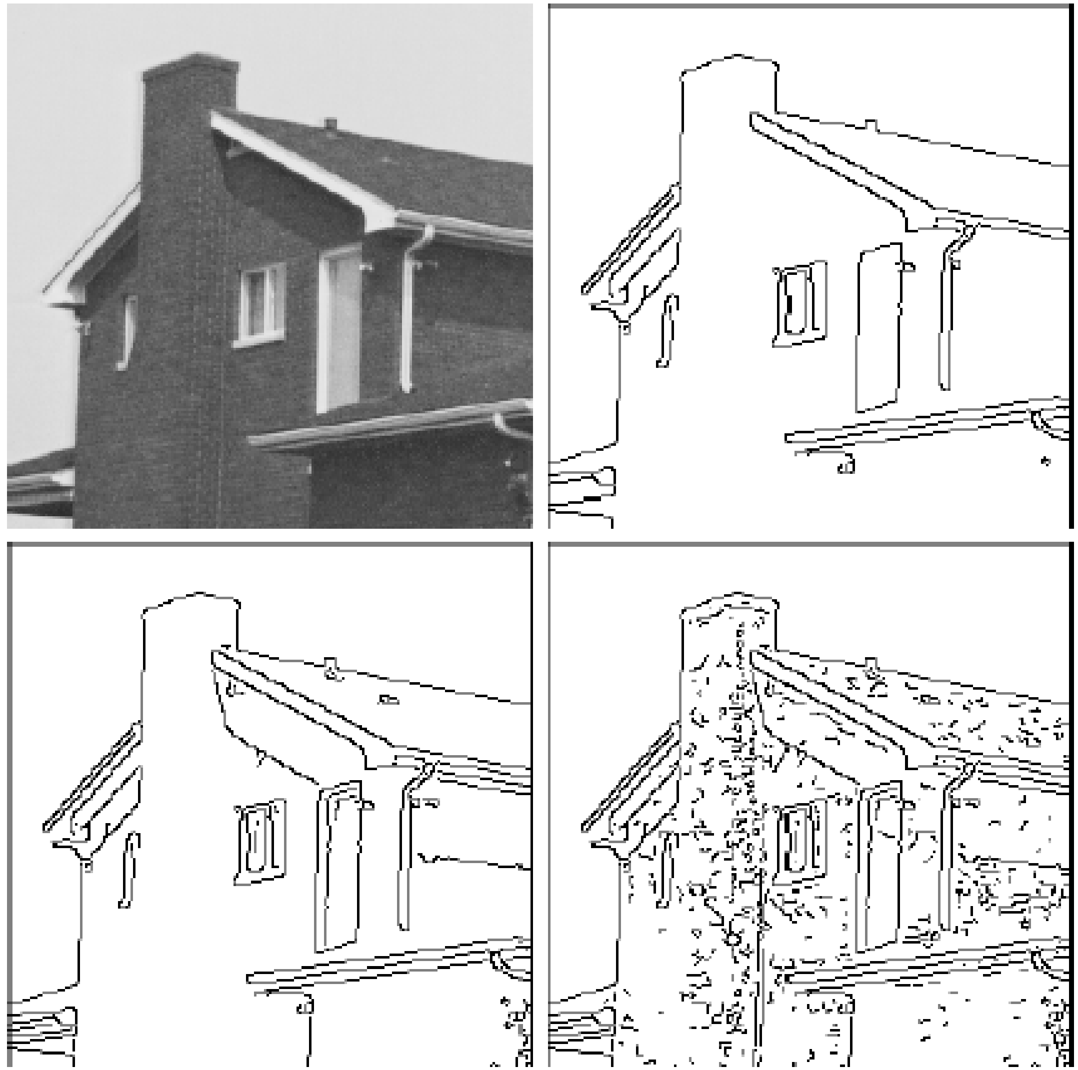
3. Hysteresis Thresholding (Double Thresholding):
   – Basic Idea: **If a weak edge lies adjacent to a strong edge, include the weak edge too**.
   – Use pixels above upper threshold T_{high} as seed points for relevant edges.
   – Recursively add all neighbours of seed points that are above the lower threshold T_{low}.

Hysteresis Thresholding

- Scan image from left to right, top to bottom
- If M(x,y) is above T_{high} mark it as edge.
- Recursively look at neighbors; if gradient magnitude is above T_{low} mark it as edge.

Hysteresis Thresholding





Influence of the upper threshold T_2 on the Canny edge detector ($\sigma = 1$, lower threshold T_1 at the 0.7 quantile). **Top left:** Original image, 256×256 pixels. **Top right:** Upper threshold at 0.95 quantile. **Bottom left:** 0.85 quantile. **Bottom right:** 0.75 quantile. Author: J. Weickert (2008).